(12) UK Patent Application (19) GB (11) 2 369 903 (13) A

(43) Date of A Publication 12.06.2002

(21) Application No 0029875.2

(22) Date of Filing 07.12.2000

(51) INT CL[7]
G06F 17/16 , H04N 1/32

(52) UK CL (Edition T)
G4A AAP

(56) Documents Cited
EP 0855681 A1

(58) Field of Search
UK CL (Edition S) G4A AAP AMV
INT CL[7] G06F 12/14 17/16 , G09C 5/00 , H04L 9/00 , H04N 1/32
ONLINE: WPI EPODOC, JAPIO, TDB, INSPEC

(71) Applicant(s)
Sony United Kingdom Limited
(Incorporated in the United Kingdom)
The Heights, Brooklands, WEYBRIDGE, Surrey, KT13 0XW, United Kingdom

(72) Inventor(s)
Jason Charles Pelly

(74) Agent and/or Address for Service
D Young & Co
21 New Fetter Lane, LONDON, EC4A 1DA, United Kingdom

(54) Abstract Title

Digital watermarking method wherein data bits are embedded in a digital material by addition or subtraction of a transforming matrix

(57) A method of embedding data in material, which material is represented by digital samples, the method comprises the steps of: defining a matrix Φ; selecting a block of samples having the same dimension as the matrix Φ; the values of the samples of the block forming a matrix of value C; and embedding a bit b of the data in the block by combining C and Φ to form a matrix Cs, wherein Cs = C+Φ represents a first value of the bit and Cs = C-Φ represents a second value of the bit. Further steps determine whether the embedded bit results in a valid matrix Cs. In some cases if the matrix would be invalid, a dummy bit may be embedded or matrix C may be clipped.

A corresponding method of decoding comprises the steps of: determining the matrix Φ; selecting a block the values of the samples in the block forming a matrix Cs, calculating Cs + Φ and Cs - Φ; calculating a function k=<Cs, Φ>; and decoding the value of the encoded bit; wherein if k>0 then the encoded bit b has the first value and if k<0 the said bit b has the second value. Further steps check whether the bit is valid.

The above steganographic ("watermarking") method is preferably applied to image data, for example acting on 8-bit image data such that a block is formed from a number of pixels and elements of the block take values in the range 0-255. The method may apply to video, audio or data material.

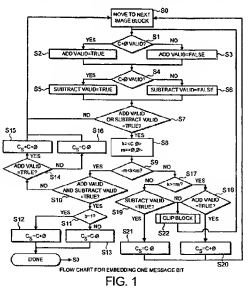
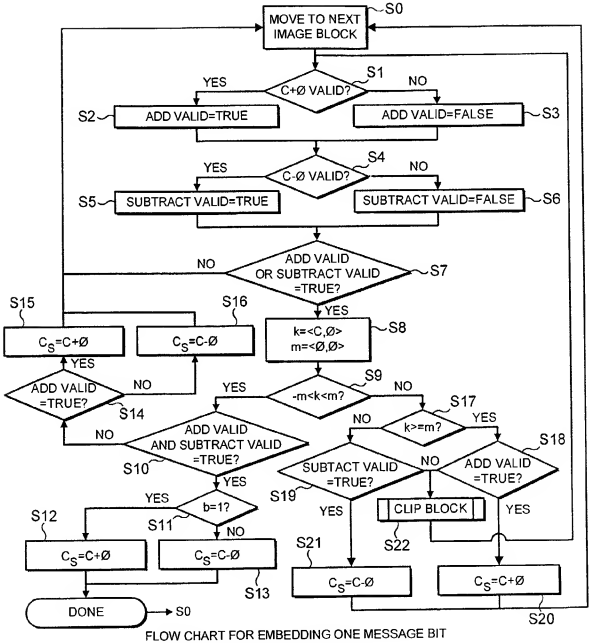


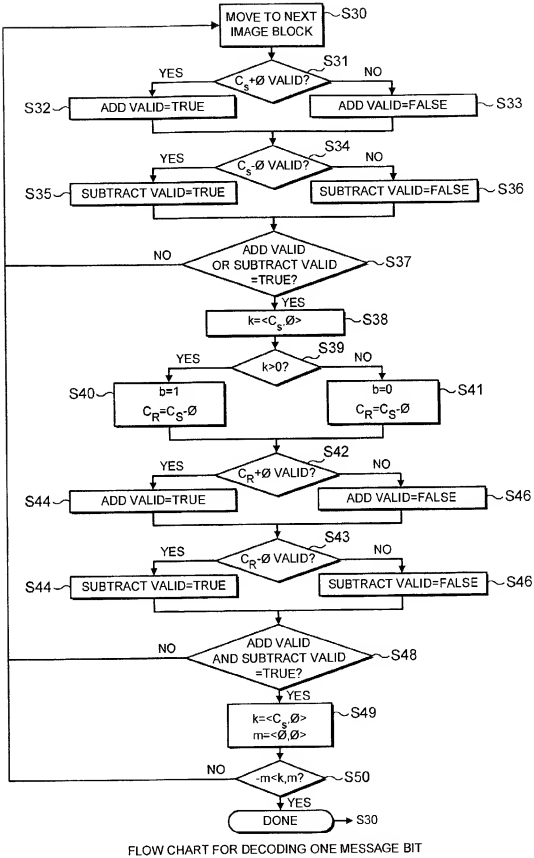
FIG. 1

GB 2 369 903 A



FLOW CHART FOR EMBEDDING ONE MESSAGE BIT

FIG. 1



FLOW CHART FOR DECODING ONE MESSAGE BIT

FIG. 2

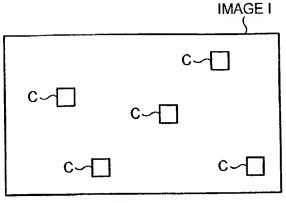


FIG. 3

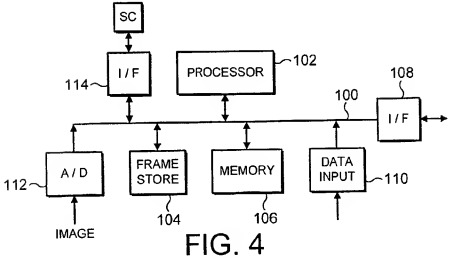


FIG. 4

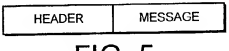


FIG. 5

2369903

Embedding Data in Material

The present invention relates to embedding data in material. Embodiments of the present invention relate to steganography.

In the embodiments of the present invention steganography is the embedding of data into material such as video material, audio material and data material in such a way that the data is imperceptible in the material.

Steganography in the form of embedding data in an image is known from for example Spread Spectrum Image Steganography by Lisa M. Marvel et al, IEEE Transactions on Image Processing Vol. 8, No 8 August 1999, pages 1075 to 1083. In this technique, random noise is modulated by data to produce a modulated noise signal, and the modulated noise signal is applied to an interleaver before adding to the image. The data may be encrypted before it modulates the noise. The interleaver reorders the signal to prevent group or burst errors in the signal.

According to one aspect of the present invention, there is provided a method of embedding data in material, which material is represented by digital samples, the method comprising the steps of:

a) defining a matrix Φ;

b) selecting a current block of samples having the same dimension as the matrix Φ, the values of the samples of the block forming a matrix of value C; and

c) combining C and Φ to form matrices

$Cs = C+\Phi$ which represents a first value of the bit and

$Cs = C-\Phi$ which represents a second value of the bit and

calculating $k = \langle C,\Phi \rangle$ and $m = \langle \Phi,\Phi \rangle$;

d) determining whether both matrices Cs are valid and $-m<k<+m$;

e) if neither matrix Cs is valid selecting the next block, and repeating steps a) to g);

f) if both matrices are valid and $-m<k<+m$, combining C and Φ to form the matrix Cs in which the desired data bit value is embedded; and

g) otherwise, either using one of Cs = C+Φ and Cs = C-Φ to embed a dummy bit, selecting the next block and repeating steps a) to g) or clipping the values of the samples in the current block and repeating steps a) to g) on the current block.

According to another aspect of the present invention, there is provided a method of decoding data bits b embedded in material by the method of said one aspect, comprising the steps of:

a) determining the value of the matrix Φ;

b) selecting a block, the values of the samples in the block forming a matrix Cs;

c) calculating Cs+Φ, Cs-Φ, m = <Φ,Φ> and k_s=<Cs, Φ>;

d) if neither Cs+Φ nor Cs-Φ are valid, move to next block and go back to (a) otherwise, if k_s>0, decoding the bit b embedded in Cs as the bit of the first value and if k_s< 0 decoding the bit embedded in Cs as the bit of second value;

e) restoring the original matrix Cr = Cs-Φ if the bit b has the first value and Cr = Cs+Φ if the bit has the second value;

f) calculating k_r=<Cr, Φ>; and

g) if –m<kr<m and both Cr -Φ and Cr+Φ are valid, b is a data bit, otherwise b is a dummy bit, so move to next block and repeat steps a) to g).

Thus data is embedded in material by altering sample values according to the value of the matrix Φ. The original values of the samples can be restored using Φ.

In a preferred embodiment, the blocks are chosen pseudo randomly. The pseudo random selection is made using a sequence of pseudo random numbers generated from a key. To decode the data the key and the matrix Φ must be available at the decoder. A smart card may store the key and matrix Φ for transfer to the decoder.

The material is preferably an image.

The invention may be used as a fragile watermark. The watermark is fragile because processing of the image, such as compression may destroy the embedded data. The invention is currently preferred to be used for embedding data in material. Large amounts of data can be embedded in an image with a wise choice of matrix Φ. Matrix Φ can have any value but in practice, the numbers forming Φ should be small

compared to the maximum value of image samples forming the matrix C. Most preferably the numbers are small compared to the average sample value of the image. Most preferably, the matrix Φ selects samples which are likely to be correlated in an image.

For a better understanding of the present invention, reference will now be made by way of example to the accompanying drawings in which:

Figure 1 is a flow diagram of a method in accordance with the invention of embedding data in an image;

Figure 2 is a flow diagram of a method in accordance with the invention of decoding and removing data embedded in an image;

Figure 3 schematically illustrates pseudo random image blocks;

Figure 4 is a schematic representation of a processor for carrying out the method of Figures 1 and 2; and

Figure 5 illustrates a data structure.

Mathematical Background

The following illustrative description assumes a basic knowledge of matrices. It refers to a matrix Φ which is a non-zero matrix and to a matrix C which has the same dimensions as Φ.

An *inner product* $\langle C, \Phi \rangle$ is defined where the inner product is

$$\langle C, \Phi \rangle = \sum_{i,j} C_{i,j} \Phi_{i,j} .$$

For example if $C = \begin{matrix} 2 & 3 \\ 4 & 5 \end{matrix}$ and $\Phi = \begin{matrix} 1 & -1 \\ -1 & 1 \end{matrix}$

then the inner product $\langle C, \Phi \rangle = 2.1 + 3.-1 + 4.-1 + 5.1 = 0$.

The inner product is only one example of a function $\langle C, \Phi \rangle$ which is usable in the present invention. Other functions are possible provided they satisfy:

(i) $\langle C + \Phi, \Phi \rangle = \langle C, \Phi \rangle + \langle \Phi, \Phi \rangle$

(ii) $\langle C - \Phi, \Phi \rangle = \langle C, \Phi \rangle - \langle \Phi, \Phi \rangle$

(iii) $\langle \Phi, \Phi \rangle > 0$ if Φ is a non-zero matrix but preferably the sum of the numbers in the matrix is zero.

The matrices C and Φ are also added. Thus for example using the same matrices as given above for C and Φ

$$C+\Phi=\begin{matrix}2+1 & 3-1\\ 4-1 & 5+1\end{matrix}.$$

Values k and m are also used where

$k = <C,\Phi>$ and

$m = <\Phi,\Phi>$.

Overview

In the following examples data is embedded in an image so as to be imperceptible in the image. The image is a digital image in which, in the following examples, each pixel is represented by an 8 bit number. A pixel has only a positive value ranging from 0 to 255. A "valid" pixel , is one having a value in that range. Such a valid pixel may be an unmodified pixel (i.e. a pixel to which no data is added) or a modified pixel (i.e. a pixel to which data is added).

The image is divided into blocks of pixels. The blocks have the same dimensions as the matrix Φ. The pixel values of the blocks form the matrix C. In preferred embodiments the blocks are a pseudo random selection within the image. The image may be a still image or a frame or field of a video image.

The matrices C and Φ may have any suitable dimensions. However the invention operates better if the pixels forming the matrix C are correlated in the image. Thus, most preferably, the pixels forming matrix C are adjacent.

.. A block or matrix is valid only if all the pixels of the block or matrix are valid.

The following description refers to "dummy" bits and "valid" bits. A valid bit is a bit encoded in a block where the following conditions are satisfied:

both C + Φ and C - Φ result in valid blocks and $-k<m<+k$.

In the following examples,

$Cs = C + \Phi$ represents a bit of value 1 and

$Cs = C - \Phi$ represents a bit of value 0.

If the said conditions are not satisfied, the decoder cannot correctly decode the bit and restore the original image. A dummy bit may be encoded in a block where the said conditions are not satisfied.

Detailed description of preferred embodiments.

The algorithm divides the image into a number of blocks with the intention that each block will be used to store one message bit. So suppose that we wish to embed one bit, b, into an image block, C.

Let Φ be a non-zero matrix with the same dimensions as C and suppose that $< C, \Phi >= 0$, where this inner product could be any appropriate matrix inner product, such as:

(1) $$< A, B >= \sum_{i,j} a_{i,j} b_{i,j}$$

Then consider the following equation that produces a stego-image block, C_s, at the encoder:

(2) $$C_s = \begin{cases} C + \Phi & \text{if } b = 1 \\ C - \Phi & \text{if } b = 0 \end{cases}$$

At the decoder, we calculate the quantity k as follows:

(3) $$k =< C_s, \Phi >$$

If $b = 1$ then $k = < C + \Phi, \Phi > = < C, \Phi > + < \Phi, \Phi > = < \Phi, \Phi >$, (*) and so k is positive. Similarly, if $b = 0$ then $k = - < \Phi, \Phi >$ which is negative. Hence by looking at the sign of k, we can determine the message bit. Knowing this, we can form the reconstructed version of C, C_r, either by subtracting Φ from C_s if we discover that $b = 1$, or by adding Φ to C_s if we discover that $b = 0$. Doing this returns the correct value of b and ensures that $C_r = C$.

It is possible to make this algorithm more general by relaxing the initial constraint of $< C, \Phi >= 0$ to be

(4) $$| < C, \Phi > | \; < \; < \Phi, \Phi >$$

There are two fundamental problems which must be overcome if this algorithm is to be practicable:

(a) assuming that Φ is fixed, we cannot guarantee to satisfy condition (4)

(b) we cannot guarantee that both $C + \Phi$ and $C - \Phi$ are valid images.

(*) because <C, Φ> = 0 as stated at (1) above.

We tackle problem (a) first. If $<C,\Phi> \geq <\Phi,\Phi>$ then we can let $C_s = C + \Phi$. This makes k positive and so at the decoder, we will deduce that $b = 1$ and that $C_r = C_s - \Phi = C$. Therefore we will be able to restore the original image. but if our original bit was a 0 then we will have decoded b incorrectly. Consequently, once the image has been restored, the decoder checks for condition (4): if it is satisfied then the bit is valid; otherwise it is a *dummy bit* and should be ignored. Similarly, if $<C,\Phi> \leq -<\Phi,\Phi>$ then we let $C_s = C - \Phi$. Again, we can reconstruct the original image correctly, but we must check for a dummy bit.

The insertion of dummy bits does increase the overall number of bits which must be embedded into the image. In general though, if Φ is chosen well then the number of dummy bits will be a small fraction of the overall payload.

Problem (b) arises from the fact that we are constraining ourselves to. say, 8-bit image data. If neither $C + \Phi$ nor $C - \Phi$ are valid images then we can leave C as it is. Then at the decoder, since Φ can be neither added nor subtracted from C_s, we know that nothing has been done to this image block.

We are now left with the situation where just one of $C + \Phi$ or $C - \Phi$ is invalid. So assume that only $C + \Phi$ is invalid. There are two situations which can arise:

(a) if $<C,\Phi> < -<\Phi,\Phi>$, then letting $C_s = C - \Phi$, we find that $k = <C - \Phi,\Phi> < 0$. We therefore detect that we need to add Φ to C_s in order to restore to C. We can then examine C to determine whether we were 'forced' to subtract Φ from C due to the invalidity of $C + \Phi$.

(b) if $<C,\Phi> \geq <\Phi,\Phi>$, then letting $C_s = C - \Phi$, gives that $k = <C - \Phi,\Phi> \geq 0$. This time we would deduce that we need to subtract Φ from C_s in order to restore to C. However, this is incorrect, and we would reconstruct not as C but as $C - 2\Phi$. Similarly, if we leave C alone then we would reconstruct as $C - \Phi$.

To solve this problem, block C is clipped so that both C+Φ and C-Φ are valid. The embedding of the data in the block is then repeated on the clipped value.

Encoding Algorithm, Figure 1.

I. At steps S1 to S6:

Check whether the image blocks $C+\Phi$ and $C-\Phi$ form valid images. If $C+\Phi$ is valid then *AddValid* is set to true; otherwise *AddValid* is set to false. If $C-\Phi$ is valid then *SubtractValid* is set to true; otherwise *SubtractValid* is set to false. If *AddValid* OR *SubtractValid*, then proceed to step III; otherwise, proceed to step II.

II. If steps S3 and S6 are both set to false indicating that both C+ Φ and C-Φ are not valid, then step S7 determines that no data can be embedded in the current image block and processing proceed to the next image block at step S0.

If at least one of steps S2 and S5 are set to true then step S7 determines that at least one value of the data bit results in valid pixel values.

Step S8 calculates k=<C, Φ> and m=< Φ, Φ>.

Step S9 determines whether –m<k<+m.

Steps S8 and S9 thus determine whether the constraint (4) above is satisfied.

III. If the constraint is satisfied and both steps S2 and S5 are set true as tested at step S10 then the data bit results in valid values whether

a) it is b=1 and Cs=C + Φ as set at steps S11 and 12; or

b) it is b=0 and Cs=C - Φ as set at steps S11 and 13.

IV. If step S10 determines that only one of steps S2 and S5 is set true, then a dummy bit is provided in the block. Step S14 tests whether AddValid was set true at step S2. If AddValid is true then Cs=C + Φ (S15) and the dummy bit is 1; and if not Cs=C- Φ (S16) and the dummy bit is 0 The procedure then moves to the next block at step S0.

V. Returning now to step S9, if -m<k<+m is not true, step S17 tests whether k>=m; that is whether <C, Φ> >=.<Φ, Φ>? If that is true, then if at step S18, AddValid is true, Cs=C + Φ (S20) and the dummy bit is 1. If k>=m is not true at step S17and if at step S19 SubtractValid is true, then Cs=C - Φ (S21) and the dummy bit is 0. The procedure then moves to the next block at step S0.

VI. If neither AddValid at step S18 nor SubtractValid at step S19 is true, then at step S22 the pixel values are clipped so as to allow both C+Φ and C-Φ to be valid. For 8 bit numbers, that is done by clipping values greater than 253 to 253 and

clipping values less than 2 to 2. The procedure then moves back to step S1 and repeats on the same block but uses the clipped values.

Decoding Algorithm, Figure 2

Φ is preset at the decoder.

I. Steps S31 to S36,operate on Cs+Φ and Cs-Φ..At steps S31 to S36

Check whether the image blocks $C_s + \Phi$ and $C_s - \Phi$ form valid images. If $C_s + \Phi$ is valid then *AddValid* is set to true; otherwise *AddValid* is set to false. If $C_s - \Phi$ is valid then *SubtractValid* is set to true; otherwise *SubtractValid* is set to false. If *AddValid* OR *SubtractValid*, then proceed to step III; otherwise, proceed to step II.

II. If steps S33 and S36 are both set to false indicating that both (Cs+ Φ) and (Cs-Φ) are not valid, then step S37 determines that no data was embedded in the current image block and processing steps to the next image block at step S30.

If at least one of steps S32 and S35 are set to true then step S37 determines that at least one value of the data bit results in valid pixel values.

Step S38 calculates k=<Cs, Φ>.

III. Step S39 tests whether k>0?

If k>0 then the bit value is 1 and the original pixel value Cr is restored as Cr = Cs - Φ. (S40).

If k<0 then the bit value is 0 and the original pixel value Cr is restored as Cr = Cs + Φ. (S41).

IV. Steps S42 to S47 test whether using the restored pixel values embed bits results in valid values i.e. whether Cr + Φ. and Cr - Φ are valid. Step S42 calculates Cr + Φ and tests whether Cr + Φ is valid (AddValid true). Step S43 calculates Cr-Φ and tests whether Cr - Φ (SubtractValid true) is valid. Steps S44, S45, S46 and S47 set flags indicating whether AddValid and SubtractValid are true.

Step S48 tests whether both AddValid and SubtractValid are true.

Step S49 calculates k=<Cr, Φ> and m=<Φ,Φ>.

Step S50 tests whether -m<k<+m.

If steps S48 and 50 are both true, then the decoded bit is a valid bit. The procedure then steps to the next block at step S30.

V. Referring to step S48, if only one of AddValid and SubtractValid for is true then the block encodes a dummy bit and the procedure steps on to the next block at stepS30.

VI. Referring to step S 50, if both of AddValid and SubtractValid for are true, but m<k<+m is not true (that is |<Cr,Φ>| >=< Φ,Φ>) then the block encodes a dummy bit and the procedure steps to the next block at stepS30.

Worked Examples

The block size for this example is 4×3 and the form of Φ is given in T3 below:

0	0	0	0
0	2	-2	0
0	0	0	0

T3 - *form of* Φ

Then $< C, \Phi > = 2(c_1 - c_2)$, where c_1 and c_2 are the image pixel values at the locations corresponding to the 2 and -2 in the block. Due to the spatial correlation of the image, we expect that c_1 and c_2 should be similar in value and hence $|< C, \Phi >|$ should be quite small. Clearly, other forms of Φ are possible, making use of the spatial correlation. Note that $< \Phi, \Phi > = 8$.

Suppose that we wish to embed a 1 and that the image block C is as in T4 below:

#	#	#	#
#	80	83	#
#	#	#	#

T4 - *initial* C

Then $|< C, \Phi >| = |2(80 - 83)| = 6$. Since we have $|< C, \Phi >| < < \Phi, \Phi >$ and it is possible to both add and subtract Φ, we set $C_s = C + \Phi$, as in T5 below:

#	#	#	#
#	82	81	#
#	#	#	#

T5 – *resultant* C_s

At the decoder, we calculate $k = < C_s, \Phi > = 2(82 - 81) = 2$. Since k is positive, we deduce we need to subtract Φ from C_s in order to restore the original block. We now have $C_r = C$. Again, we have $|< C_r, \Phi >| < < \Phi, \Phi >$ and it is possible to both add and subtract Φ. Therefore we deduce that the bit is a message bit and not a dummy bit. Since k is positive, the message bit is a 1.

2. For the same block size as in T4 and for the same value of Φ as given in T3, Φ=[2, -2], assume C=[3, 1] and the bit b to be embedded is b=1. Then referring to steps S1 to S6 in Figure 1, C+Φ=[5, -1] which is invalid, and C-Φ=[1, 3] which is valid.

At step S7 only one of AddValid and Subtract Valid is true.

At step S8, k=6 - 2=4 and m= 4 + 4 =8.

Thus at step S9 $-m<k<m$ ($-8<4<8$). But at step S10 only one of AddValid and Subtract Valid is true so a bit is coded as a dummy bit b=0 at step S16 because only SubtractValid is true. Thus Cs=[1, 3]

Referring to Figure 2, steps S31 and S34, calculate Cs+Φ and Cs-Φ. Φ=[2,-2]

Thus Cs+Φ=[3, 1] and Cs -Φ =[-1, 1]. Thus only AddValid for Cs + Φ is true in steps S32 to S37.

At step S38, k=<Cs, Φ> = 2 - 6 =-4 is negative at Step S39 so at step S41 b=0 and Cr = Cs + Φ = [3,1].

Steps S 42 to S48 show that Cr + Φ is invalid and Cr - Φ is valid and so the bit is a dummy bit. The decoding process thus steps to step S30 for the next block.

3. Assume Φ = [2, - 2] and C = [255,250].

Referring to Figure 1,steps S1 to S6, C + Φ = [257, 248] which is invalid because 257 is greater than the largest possible pixel value of 255 for 8 bit pixels but C - Φ= [253, 252] which is valid Therefore only one of AddValid and Subtract Valid is true at step S7.

At step S8, k= 510 - 500= 10 and m = 8.

At steps S9 and S17 $k \geq m$. At step S18 AddValid (C + Φ) is not true, so at step S22 the pixel value of 255 is clipped and the process returns to step S1 and repeats on the same block but operates on the clipped value which may result in valid encoding of a bit in the block.

4. Assume Φ = [2, - 2] and C = [100,200].

Referring to Figure 1,steps S1 to S6, C + Φ = [102, 198] which is valid.

C - Φ= [98, 202] which is valid. Therefore at least one of AddValid and Subtract Valid is true at step S7.

At step S8, k= 200 - 400= -200 and m = 8.

At steps S9 and S17 k ≤-m. At step S18 AddValid (C + Φ) is true, so at step S20 Cs= (C + Φ) = [102, 198] encoding dummy bit b=1 and then the process steps to step S0 for the next block.

Referring to Figure 2, at steps S31 to S37, Cs + Φ = [102, 198] which is valid and Cs - Φ=[98, 202] which is also valid. So, at least one of AddValid and SubtractValid is true at step S37. At step S38, k= 200-400 = - 200. At step S39 k<0 so bit b=1 (S40) and Cr = [100, 200]. Steps S42 and S43 reveal that Cr + Φ =[102, 198] is valid and (Cr - Φ) = [98, 202] is also valid. However, steps 49 and 50 show k< - m so the bit is a dummy bit and the procedure steps to step S30 for the next block.

In preferred embodiments of the invention, the blocks C to be encoded are chosen at random throughout an image as shown in Figure 3. Steps S0 and S30 then comprises selecting the next block in a pseudo-random sequence of blocks. The sequence may be defined in known manner using a key, which defines the pseudo-random sequence. That key, like Φ, is available at both the encoder and the decoder.

Illustrative system.

Referring to Figure 4 a system for encoding comprises a data bus 100 to which there are coupled a processor 102, a frame store 104 for storing a digital image, other system memory 106, an interface 108, a data input device 110 and an analog to digital converter 112. An analogue image is digitised by the A/D converter 112 and stored in the frame store 104. The processor operates according to Figure 1 to embed data in the image. The data may be a message entered via the input device 110 which is e.g. a keyboard. The encoded image may be output to a transmission channel via the interface 108. The processor determines the selection of blocks using a sequence of pseudo random numbers generated from a key. Also the processor determines the values in the matrix Φ. The key and matrix Φ are needed to decode the embedded data and to restore the original image. Thus for example, the encoder has a smart card

interface 114 for receiving a smart card SC. The key and matrix Φ are stored in the card for transfer to the decoder.

The decoder may have the same structure as the encoder, receiving an encoded image via the interface 108, and the key and matrix Φ from a smart card SC inserted into the interface 114. The processor 102 operates according to Figure 2. The image is stored in the frame store for that purpose. The A/D converter 112 of the encoder may be replaced by a D/A converter in the decoder.

Modifications.

a) As in T3, a border can be left around the non-zero elements of Φ. The corresponding image block pixels will therefore be unaffected by the steganographic process. It is therefore possible to use them for measuring the activity in the block and Φ can be scaled accordingly – the larger the activity, the larger the values of Φ. This has two distinct advantages, namely (i) with large activity, the value of $|< C, \Phi >|$ is likely to be larger, so a larger Φ will be required for condition (4) to be satisfied and (ii) with larger activity, larger pixel changes can be made whilst maintaining visual quality.

b) It is possible to make use of the above border in another way. In order to avoid boundary problems, where either $C + \Phi$ or $C - \Phi$ are invalid, the border pixels can be examined. If any of them take too large or too small a value, then it is possible that some of the changed pixels may also take too large or too small a value once Φ is 'applied', causing a boundary problem. On the other hand, if the border pixels take 'average' values then the pixels to be altered are likely to assume 'average' values too. Thus by checking border pixels for 'near-extreme' values, the likelihood that a boundary problem occurs is reduced.

c) It is possible to insert headers into the message, each header specifying the number of message bits which are encoded prior to the next occurrence of a boundary problem. In this way, all boundary problems are concentrated on the locations where the header bits are recorded. Although this does increase the overall number of bits to be embedded, the likelihood of lossless image reconstruction is increased.

The data to be embedded may include a data structure as outlined in Figure 5. The header contains h bits where h is 8 for example. Blocks are selected for the embedding of data as described above. The first set of h blocks in which valid data bits are embeddable are assigned to the header data. Subsequent blocks are selected and examined to determine if valid data bits can be embedded. A block which cannot contain a valid data bit is skipped. The header contains a number indicating the number of the skipped blocks counting from the first block after the header. After the skipped block, a new header is provided over the next h blocks for indicating the next skipped block.

To encode and decode messages, the value of Φ and the key to the pseudo-random selection of blocks are required. Both need to be available at the encoder and the decoder. They provide security against unauthorised decoding. The value of Φ and then key may be stored in e.g. a smart card which is secure and the processor 102 of Figure 4 accesses the key and Φ on the smart card via a suitable interface in known manner. Other secure ways of transferring the key and matrix Φ to the decoder may be used; for example they may be transferred in encrypted form over a communications network.

If a message has a greater number of bits than can be encoded in the number of valid blocks available in an image then the bits which are not encoded in the blocks may be encoded in pixels using the techniques of altering least significant bits or some other known suitable technique.

Whilst the embodiments of the invention described herein refer to matrices or blocks of dimension 1x2, other dimensions may be used.

For example, matrix Φ may be :

a) $\begin{matrix} 1 & 0 \\ 0 & -1 \end{matrix}$ or b) $\begin{matrix} 0 & 0 & 0 & 0 \\ 0 & 2 & -2 & 0 \\ 0 & 0 & 0 & 0 \end{matrix}$ or c) $\begin{matrix} 3 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & -3 \end{matrix}$. Those are merely examples. Matrix c) is not a good choice because the non zero numbers which select image pixels are far apart and unlikely to be correlated. However such a matrix could be used in principle. Blocks such a b) which have zero borders may have overlapping borders provided the non-zero values do not overlap.

Whilst the embodiments of the invention described herein refer to images, the invention may be applied to audio material, and other data material. Image material may be still or moving images.

CLAIMS

1. A method of embedding data in material, which material is represented by digital samples, the method comprising the steps of:

a) defining a matrix Φ;

b) selecting a current block of samples having the same dimension as the matrix Φ, the values of the samples of the block forming a matrix of value C; and

c) combining C and Φ to form matrices

$Cs = C+\Phi$ which represents a first value of the bit and

$Cs = C-\Phi$ which represents a second value of the bit and

calculating $k = \langle C,\Phi \rangle$ and $m = \langle \Phi,\Phi \rangle$;

d) determining whether both matrices Cs are valid and $-m<k<+m$;

e) if neither matrix Cs is valid selecting the next block, and repeating steps a) to g);

f) if both matrices are valid and $-m<k<+m$, combining C and Φ to form the matrix Cs in which the desired data bit value is embedded; and

g) otherwise, either using one of $Cs = C+\Phi$ and $Cs = C-\Phi$ to embed a dummy bit, selecting the next block and repeating steps a) to g) or clipping the values of the samples in the current block and repeating steps a) to g) on the current block.

2. A method according to claim 1, wherein if $-m<k<+m$ is not true and at least one of $C+\Phi$ and $C-\Phi$ is true, the samples of the current block are clipped if a dummy bit is not valid in the block.

3. A method according to claim 2, wherein $k>=m$ is tested and if $k>=m$ is true then $C+\Phi$ is embedded as a dummy bit if it is valid and if $k>=m$ is not true then $C-\Phi$ is embedded as a dummy bit if it is valid.

4. A method according to claim 1, 2 or 3 wherein the blocks are selected pseudo randomly.

5. A method according to claim 1, 2, 3 or 4 wherein the said material is an image.

6. A method according to any preceding claim, comprising the step of storing the matrix Φ.

7. A method according to any preceding claim, wherein the data bits b are of a data structure having a header section denoting the number of message bits and a value section containing the message bits.

8. A method of decoding data bits b embedded in material by the method of anyone of claims 1 to 7, comprising the steps of:

a) determining the value of the matrix Φ;

b) selecting a block, the values of the samples in the block forming a matrix Cs;

c) calculating $Cs+\Phi$, $Cs-\Phi$, $m = <\Phi,\Phi>$ and $k_s=<Cs, \Phi>$;

d) if neither $Cs+\Phi$ nor $Cs-\Phi$ are valid, move to next block and go back to (a) otherwise, if $k_s>0$, decoding the bit b embedded in Cs as the bit of the first value and if $k_s< 0$ decoding the bit embedded in Cs as the bit of second value;

e) restoring the original matrix $Cr = Cs-\Phi$ if the bit b has the first value and $Cr = Cs+\Phi$ if the bit has the second value;

f) calculating $k_r=<Cr, \Phi>$; and

g) if $-m<kr<m$ and both $Cr -\Phi$ and $Cr+\Phi$ are valid, b is a data bit, otherwise b is a dummy bit, so move to next block and repeat steps a) to g).

9. A method according to claim 8, wherein the value of the matrix Φ is determined from a stored value thereof.

10 A method according to claim 8 or 9, comprising calculating

$Cr = Cs+\Phi$ or $Cs-\Phi$

depending on the decoded bit value to restore the original value.

11. A decoding method according to any one of claims 8 to 10, wherein if $k=0$ then the data bit is decoded as one of the first and second values.

12. An embedding method according to any one of claims 1 to 7, wherein the data bits are in a data structure comprising a header section and a message section.

13. An embedding method according to claim 12, wherein the header identifies the next block after the header which is not used.

14. A computer program product arranged to carry out the method of any preceding claim when run on a computer.

15. Encoding apparatus arranged to carry out the method of any one of claims 1 to 7,12 and 13.

16. Decoding apparatus arranged to carry out the method of any one of claims 8 to 11.

17. A method substantially as hereinbefore described with reference to the accompanying drawings.

18. Apparatus substantially as hereinbefore described with reference to the accompanying drawings.

19. A memory device for use with the encoding apparatus of claim 15 and/or the decoding apparatus of claim16 the device securely storing the matrix Φ.

20. A device according to claim19 further storing a key for generating a pseudo random number sequence for selecting blocks.

21. A device according to claim 19 or 20 which is a smart card.

Application No:	GB 0029875.2	**Examiner:**	Paul Jefferies
Claims searched:	1-21	**Date of search:**	26 November 2001

Patents Act 1977
Search Report under Section 17

Databases searched:

UK Patent Office collections, including GB, EP, WO & US patent specifications, in:

UK Cl (Ed.S): G4A (AAP, AMV)

Int Cl (Ed.7): G06F 12/14, 17/16; G09C 5/..; H04N 1/32; H04L 9/..;

Other: ONLINE: WPI, EPODOC, JAPIO, TDB, INSPEC

Documents considered to be relevant:

Category	Identity of document and relevant passage	Relevant to claims
A	EP 0855681 A2 (NIPPON) See e.g. page 21, line 38 et seq. and figure 30	

X	Document indicating lack of novelty or inventive step	A	Document indicating technological background and/or state of the art.
Y	Document indicating lack of inventive step if combined with one or more other documents of same category.	P	Document published on or after the declared priority date but before the filing date of this invention.
&	Member of the same patent family	E	Patent document published on or after, but with priority date earlier than, the filing date of this application.